Amendments to the Claims

Please amend claims 1, 2, 5, 6, 9, 10 and 13. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1. (Currently Amended) A method for updating a lookup table comprising the steps of:

providing access to a first set of routes stored in nodes of a first subtree, ~~and associated first subtree entry stored in a first memory space in the lookup table~~ the first subtree being accessed through a first pointer to [[the]] a first subtree ~~entry~~ root node; ~~and~~

storing a second set of routes stored in nodes of a second subtree, ~~and associated second subtree entry in a second memory space in the lookup table~~ the second subtree being accessed through a second pointer to a second subtree root node, while access is provided to the first set of routes stored in the first subtree ~~memory space~~ by the first pointer; and

switching access to the second set of routes stored in the second subtree ~~memory~~ by replacing the first pointer ~~stored~~ to the first subtree ~~entry~~ root node with [[a]] the second pointer to the second subtree ~~entry~~ root node.

2. (Currently Amended) The method as claimed in Claim 1 further comprising the step of:

deallocating ~~the first~~ memory ~~space~~ used by the first set of routes after switching access.

3. (Original) The method as claimed in Claim 1 wherein the number of routes in the first set of routes is less than the number of routes in the second set of routes.

4. (Original) The method as claimed in Claim 1 wherein the number of routes in the first set of routes is greater than the number of routes in the second set of routes.

5. (Currently Amended) An apparatus for updating a lookup table comprising:

a first pointer to a first subtree ~~entry~~ root node, the first subtree ~~entry~~ root node providing access to a first set of routes stored in ~~a first memory space~~ nodes of a first subtree; and ~~means for~~

a second pointer to a second subtree root node, the second subtree root node providing access to a second set of routes stored in nodes of a second subtree, ~~storing a second set of routes and associated second subtree entry in a second memory space~~ while access is provided to the first set of routes ~~stored in the first memory space~~ by the first pointer and switching access to the second set of routes by replacing the first pointer to the first subtree root node with [[a]] the second pointer to the second subtree ~~entry~~ root node~~, the second subtree entry providing access to the second memory space~~.

6. (Currently Amended) The apparatus as claimed in Claim 5 further comprising:

means for deallocating ~~the first~~ memory used by the first set of routes ~~space~~ after switching access.

7. (Original) The apparatus as claimed in Claim 5 wherein the number of routes in the first set of routes is less than the number of routes in the second set of routes.

8. (Original) The apparatus as claimed in Claim 5 wherein the number of routes in the first set of routes is greater than the number of routes in the second set of routes.

9. (Currently Amended) An apparatus for updating a lookup table comprising:

a first pointer to a first subtree ~~entry~~ root node, the first subtree ~~entry~~ root node providing access to a first set of routes stored in nodes of a first subtree ~~a first memory space~~;

~~a second memory space for storing~~ a second set of routes stored in nodes of a second subtree, the second subtree being accessed by a second pointer, ~~and associated~~

~~second subtree entry~~ while access is provided to the first set of routes stored in the first ~~memory space~~ subtree by the first pointer; and

logic which provides access to the second set of routes by replacing the first pointer to the first subtree root node with [[a]] the second pointer to the second subtree ~~entry~~ root node~~, the second subtree entry providing access to the second memory space after the second set of routes are stored in the second memory~~.

10. (Currently Amended) The apparatus as claimed in Claim 9 further comprising:

deallocation logic which deallocates the ~~first~~ memory ~~space~~ used by the first set of routes after the first pointer is replaced.

11. (Original) The apparatus as claimed in Claim 9 wherein the number of routes in the first set of routes is less than the number of routes in the second set of routes.

12. (Original) The apparatus as claimed in Claim 9 wherein the number of routes in the first set of routes is greater than the number of routes in the second set of routes.

13. (Currently Amended) A method for updating a lookup table, the lookup table providing a longest prefix match for a destination address, comprising the steps of:

providing access to a first set of routes stored in nodes of a ~~and associated~~ first subtree, the first subtree being accessed ~~entry stored in a first memory space in the lookup table~~ through a first pointer to [[the]] a first subtree ~~entry~~ root node; ~~and~~

storing a second set of routes stored in nodes of a second subtree, the second subtree being accessed through a second pointer to a second subtree root node, ~~and associated second subtree entry in a second memory space in the lookup table~~ while access is provided to the first set of routes stored in the first subtree ~~memory space~~ by the first pointer; and

switching access to the second set of routes stored in the second subtree ~~memory~~ by replacing the first pointer ~~stored~~ to the first subtree ~~entry~~ root node with [[a]] the second pointer to the second subtree ~~entry~~ root node.

14. (Previously Presented) The method of claim 13, wherein the first set of routes and the second set of routes include a longest prefix route for the destination address.

15. (Previously Presented) The method of claim 14, wherein the destination address includes an Internet Protocol address.

16. (Previously Presented) The method of claim 14, wherein the second set of routes includes another route corresponding to the longest prefix route for another destination address.

17. (Previously Presented) The method of claim 13, wherein the first set of routes and the second set of routes are associated with nodes at the bottom level of a subtree.